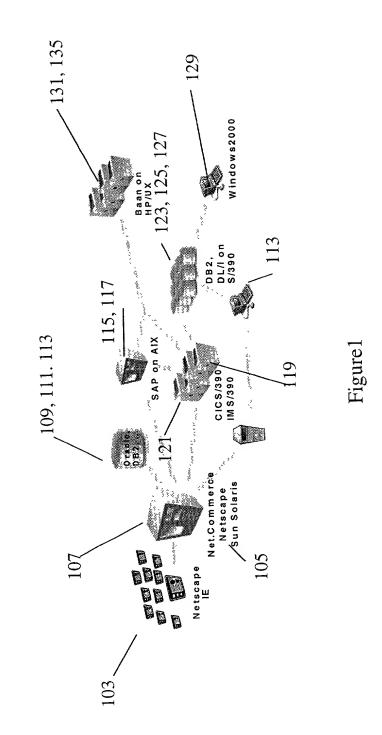
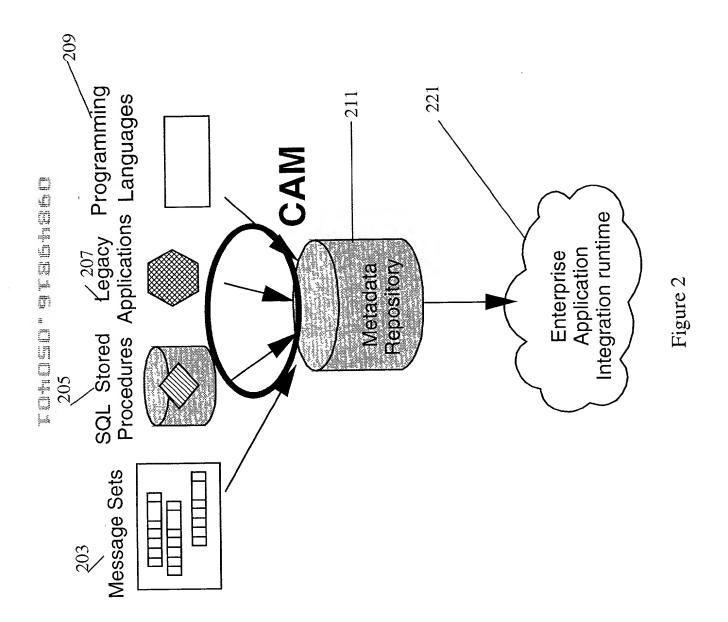
101





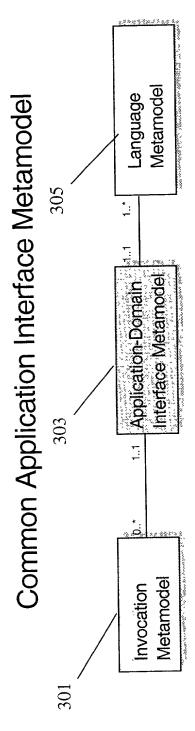
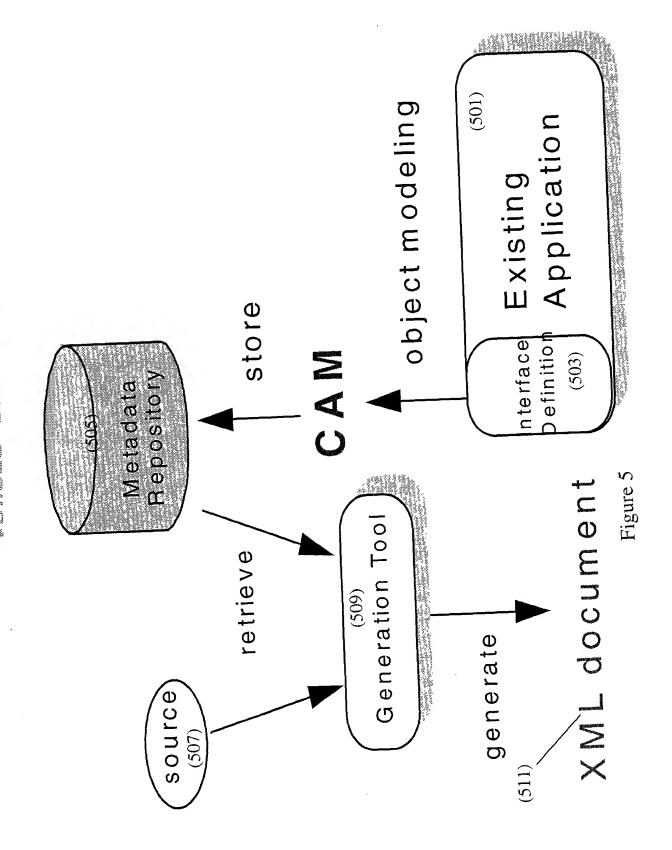
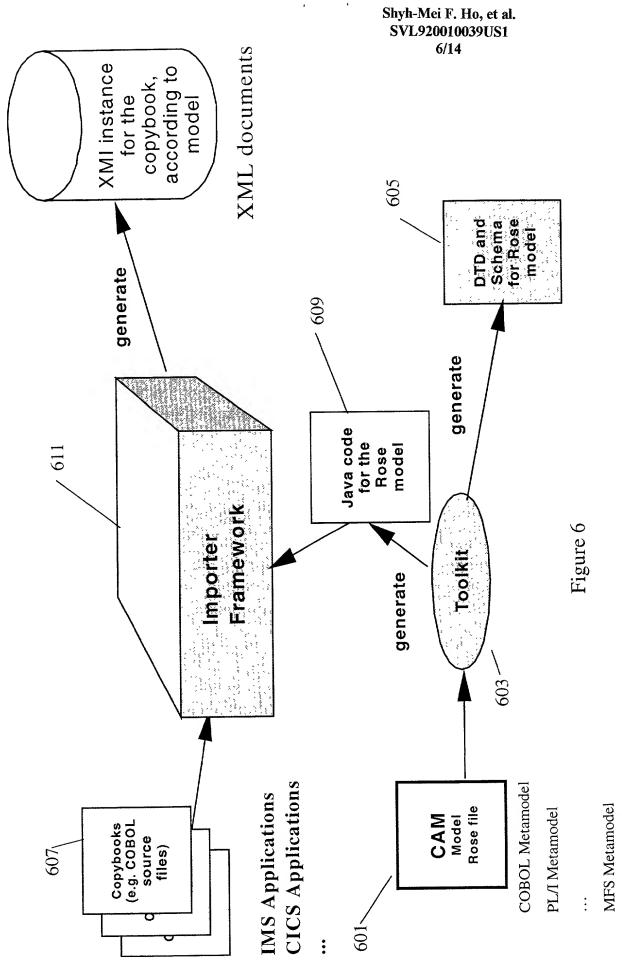


Figure 3

C Metamodel 427 Metamodel Figure 4 425 Transaction Metamodel Message 423 OTMA Invocation Metamodel 421





BMS Metamodel

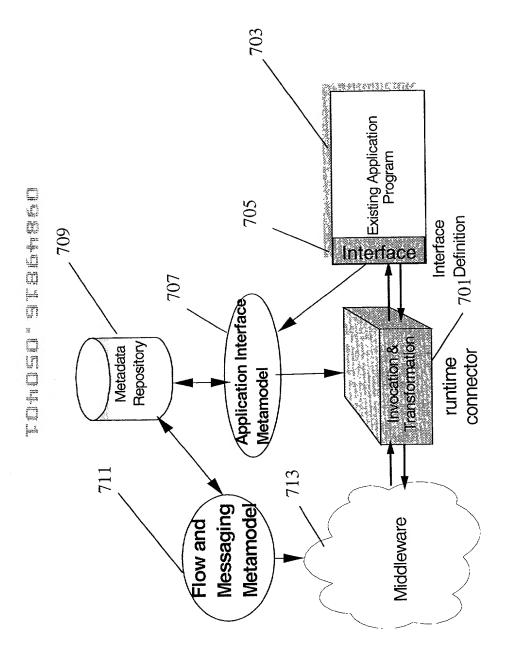
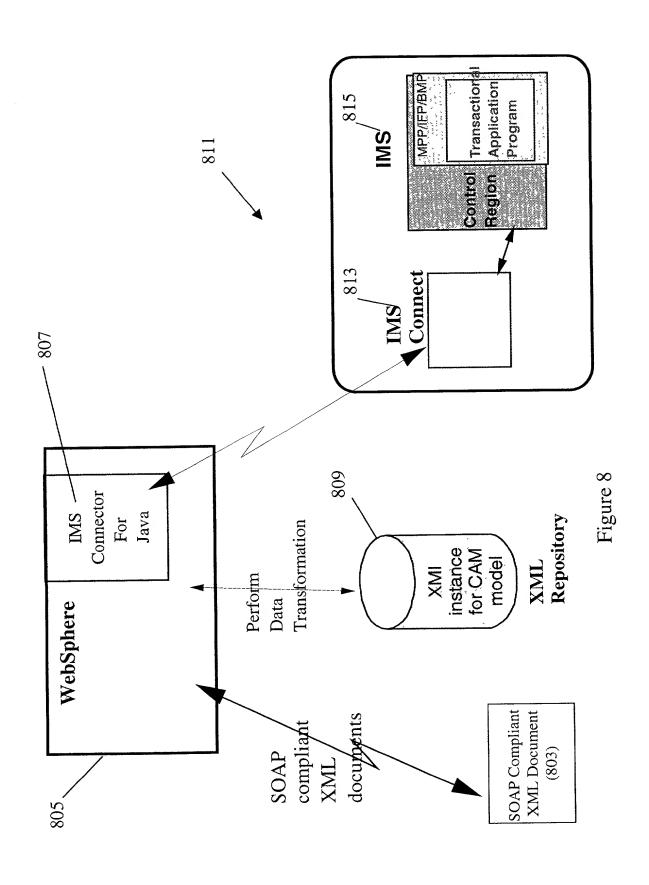


Figure 7



Shyh-Mei F. Ho, et al. SVL920010039US1 9/14

ø			+M essage Contain er		+MessageComponent	1 1 Application Data (from Application Data)	+ Field Container	• -	т. Б				6
<pre><eenumeration>> OT M AP refix Formats ormats = one</eenumeration></pre>		+StandardFleldsContainer		+StandardFieldsComponent 0 1	StandardFields	StandardFields Length TwoByteField PARservedField TwoByteField TransactionCode VariableLengthField	{The Transaction Code field can be from 1 to 8 bytes in length lits included only in input messages }		In direct XML-to-IM S application scenarios. Standard Fields are not included				Figure 9
IM STransaction Message «Format OTM A Prefix Formats = one sids Flag boolean	*	+		+Star		A ength	C	+UserDataContainer	+UserDataComponent	0.1	UserData	{ U se rData op to naly appears in the pre first segment the first segment of all messages }	
IM ST ran. OT M A PrefixForm at Standard FieldsFlag			+OTMAPrefixContainer		. 0	OTMAPrefix 1.1	e curity Data Contair	1+	+S tateDataComponent	+SecurityDataComponent 0 1	Security Data	{SecurityData optionaly appears in the prefix preceding the first segment of all messages }	Format "one": a prefix appears before all message segments Format "two" a prefix appearsonly before the first segment.
							e Data Contai			DataComp 1.1	1.1 State Data	State Data appears in the prefix preceding the first segment of all messages	Format "one", a prefix ap before all message segm Format "two" a prefix ap before the first segment.
							+C on tro ID at a C on 1a mer + S ta t			+Contro ID at a Component	ControlData	{ Control Data appears in all prefixes. }	

+ControlDataContainer

+StateDataContainer

+Security Data Container

+Use rDataContainer

+ControlDataComponent

ControlData

-

Architecture Level: One Byte Field Length. Two Byte Field Secont yellogh: Two Byte Field Secont yellogical Synchronization Flag. The second Secont yellogical Synchronization Level Towns of Secont yellogical Synchronization Level Towns of Secont Yellogical Secont Yellogical

CommandType: TCommandType ProcessingFlag: TProcessingFlag TpipeName: EightByteField ChainFlag: TChainFlag

Shyh-Mei F. Ho, et al. SVL920010039US1 10/14

UserData: VanableLengthField

Length: TwoByteField User Da ta

UserIDL ength: OneByteField

UserIDT ype: On eB yteField

ServerToken : SixteenByteField CorrelatorToken : SixteenByteField

MapName: EightByteField Reserved : OneByteField

ContextID: SixteenByteField

Utoken: Variable Length Field

Utoken Type: On eBy teFle Id

Profile Length: OneByteField Profile: Variable Length Field

UserID: Variable Length Field Profile Type: On eByteField

SendSequenceNumber: FourByteField PrefixFlag: TPrefixFlag

Recoverable Sequence Number Four Byte Field ReasonCode: TwoByteField SenseCode: TwoByteField

ServerUserDataLength: TwoByteField ServerUserData: VariableLengthField DestinationOverride . EightByteField

> SegmentSequenceNumber: TwoByteField Reserved: TwoByteField

Figure 10

+SecurityDataComponent

+StateDataComponent

State Data

SecurityData

Shyh-Mei F. Ho, et al. SVL920010039US1 11/14

<<enumeration>>

<<enumeration>> TMessageType

Data: String Transaction: String Response: String

TCommitConfirmationFlag Committed: String Aborted: String

<<enumeration>> **TCommandType**

ClientBid : String ServerAvailable : String CBresynch: String

SuspendProcessingForAllTpipes: String

ResumeProcessingForAllTpipes: String

Command: String CommitConfirmation: String

SuspendInputForTpipe: String ResumeInputForTpipe: String ResumeInputForTpipe: String Resynch: String

REQresynch : String REPresynch: String

TBresynch: String

<<enumeration>> TPrefixFlag

> <<enumeration>> TChainFlag

<<enumeration>> **TProcessingFlag**

Security Data: String

♥UserData: String ♥ApplicationData: String

StateData: String

Figure 11

MiddleInChain: String DiscardChain: String LastInChain: String FirstInChain: String

AsynchronousOutput: String ErrorMessageFollows: String SynchronizedTpipe: String

<<enumeration>> **TSeverState** Conversational State: String Responsel/Mode: String

TSynchronizationFlag <<enumeration>>

CommitThenSend: String SendThenCommit: String

TSynchronizationLevel <<enumeration>>

None: String Confirm: String SYNCPT: String

Figure 12

<<enumeration>> TSecurityFlag

NoSecurity : String Check : String Full : String

Figure 13

Shyh-Mei F. Ho, et al. SVL920010039US1 14/14

	Note: these primitives don't tell you the	format of the information, just the	byte-lengths. We may need to modify	this if we want to specify that the data is,	for example, two bytes of ASCII versus	two bytes of Unicode, or that the bytes	are big-endian ordered versus little-endian.
< <pre><<pre><<pre><<pre><<pre><<pre></pre></pre></pre></pre></pre></pre>	OneByteField					< <p>< </p>	naphiela

< <pre><<pre><<pre><<pre>SixteenByteField</pre></pre></pre></pre>	< <pre><<pre><<pre>/ariableLengthField</pre></pre></pre>
< <pre><<pre><<pre><<pre>EightByteField</pre></pre></pre></pre>	Var
< <pre><<pre>c<pre>primitive>> FourByteField</pre></pre></pre>	< <pre><<pre>c<pre>c<pre>primitive>> SixByteField</pre></pre></pre></pre>

Figure 14